

基于以太网的嵌入式源代码级调试系统设计

朱旭峰, 周剑扬, 戴祖彬

(厦门大学 信息科学与技术学院 福建 厦门 361005)

【摘要】: 随着嵌入式技术的迅速发展, 一个好的交叉调试器对缩短嵌入式软件的开发周期有着重要意义。本文针对 ARC810 处理器的现有调试方式下载速度慢的不足, 提出了通过 PCI 以太网卡下载待调试程序的调试方法。经过实验, 下载速度达到 418kB/s, 满足了快速下载的需要, 并实现了源代码级调试。

【关键词】: ARC810 交叉调试器 EDCL

0、引言

在嵌入式软件开发的过程中, 由于目标机通常不具备自举开发的能力, 嵌入式软件开发大多采取远程调试(交叉调试)的方式, 即让调试器运行在宿主机上, 被调试程序运行在目标机上, 主机和目标机之间通过某种通信端口如串口、并口、以太网等进行连接。

ARC810 是台湾信亿科技生产的 32 位处理器, 笔者所在的实验室为其开发了一套调试工具^[1], 并命名为 ARCDU, 该调试工具的宿主为 PC 机, 通过串口发送调试命令给目标机, 待调试的程序也通过串口进行下载。经过测试, 发现下载速度只有 3.79kB/s, 当调试比较大的程序, 如 Linux 内核, 下载速度慢影响了调试速度。针对这个不足, 本文根据 ARC810 具有 PCI 接口的特点, 设计了通过 PCI 以太网卡下载待调试程序的方案, 下载速度达到 418kB/s, 取得了满意的结果。

1、ARC810 调试基本原理

ARC810 处理器是基于 LEON2 内核的一款 SOC 芯片, LEON2 内核是欧洲航天局所属研究所开发的基于 SPARC V8 架构的开源的 32 位 RISC 处理器。ARC810 的调试方法与 LEON2 相同, 通过芯片上的调试支持单元进行软件调试, 该单元由 2 个模块组成: 调试支持单元(DSU)和调试通信链路(DCL)。DSU 能够使处理器进入调试模式, 只有在调试模式下才能访问处理器的寄存器和高速缓存器, DSU 还包含了一个跟踪缓存器, 这个缓存器存储了执行过的指令和在 AHB 总线上传输的数据。DCL 使用一个标准的异步 UART 来实现与宿主机上运行的软件调试器通信, 通信协议是简单的读写协议^[2]。该硬件调试单元的模块图如图 1 所示。

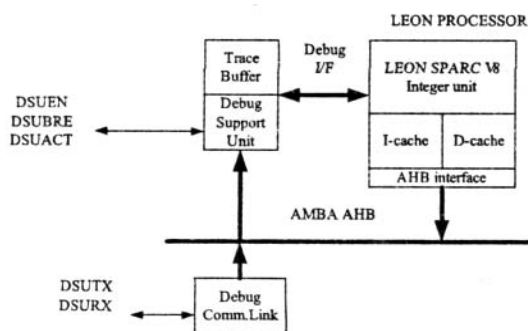


图 1 LEON2 硬件调试单元模块图

笔者所在的实验室以模块化的思想为 ARC810 开发了一套调试工具 ARCDU, 其宿主机和目标机之间采取的串口通讯方式极大的限制了待调试程序的下载速度, 因此本文设计了通过以太网卡下载待调试程序的方案并加以实现。

2、基于以太网调试的方案设计

2.1 硬件系统结构

ARC810 的内部模块图^[3]如图 2 所示:

该处理器采用 SPARC V8 结构的 32 位处理器, 具有分开的指令和数据缓冲, 硬件乘除法器, 中断控制器, 带有跟踪缓冲区

的调试支持单元、2 个 24 位 Timer、2 个 UART、1 个 watchdog、16 位的 I/O 口、内存控制器、PCI 接口和 SPI。

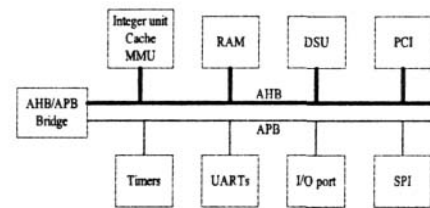


图 2 ARC810 内部模块图

ARC810 支持 PCI2.2 标准, 可以通过 PCI 口扩展外设。本文的实验平台具有 PCI 插槽, 在插槽上插入 D-Link 公司所生产的 DFE-530TX, REV-C2 网卡, 通过该 PCI 网卡实现目标机和宿主主机之间的通讯。

2.2 软件系统结构

2.2.1 目标机网卡监听程序

该网卡监听程序由于涉及到以太网数据的接受和发送, 因此和网卡的驱动程序在网卡初始化, 网卡数据缓冲区的设计上都有相似之处, 但是由于并不和操作系统打交道, 因此在数据包的接受和发送上面与驱动程序又有所不同。系统流程图如图 3 所示:

网卡的初始化主要完成网卡的挂载, 网卡相关寄存器的初始化, 网卡发送和接受缓冲区的分配等工作。只有网卡正常初始化之后, 数据的接受和发送才能正常的进行。在网卡的初始化过程中由于要对网卡寄存器进行读写, 因此首先要确保 I/O 函数的正确性, 尤其要注意大小端的问题, 在分配发送、接受缓冲区的时候要根据所用网卡的硬件要求来处理字节对齐的问题^[4], 否则网卡不能正确的收发。由于不涉及到操作系统, 我们还需要读出网卡的 MAC 地址, 以供网卡在接收和发送数据时使用。

网卡采取轮询的方式来接收 PC 机发送的数据包, 并根据数据包的种类回应相应的包。由于没有网络协议栈, 收到的数据包是否是目标机所需要的, 以及数据包的类型都需要我们自己来判断。判断的依据是数据包的以太网包头。我们根据包头的 type code 来判断收到的数据包是 ARP 包还是 IP 包。如果是 ARP 包, 再进行 IP 地址匹配检测之后决定是否回应 ARP 包; 如果是 IP 包, 还要判断此 IP 包是否是需要的 UDP 包。

在本监听程序的设计当中, 网卡不主动向外发送数据, 只有当网卡接收到数据包之后, 再根据数据包的类型来决定是否回应此数据包。在回应 ARP 包的时候, 按照 ARP 包格式进行组包, 在回应 UDP 包的时候, 按照 UDP 包格式进行组包。其中要

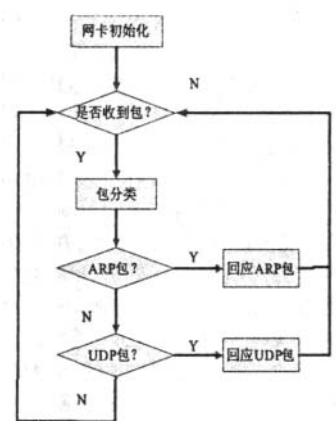


图 3 目标机网卡监听程序流程图

注意的是,由于没有网络协议栈,因此以太网头和 IP 包头都需要自行添加。在进行 UDP 的组包的时候,IP 头中的 header checksum 域要自行计算,如果此域填写不正确,那么将不能正常的和宿主机进行通信。

2.2.2: 宿主机数据发送程序

宿主机端的程序首先将待下载程序文件等长度分包,然后根据通信协议与目标机进行交互。宿主机端要维护一个全局的序列号,在发送数据包的时候,当前序列号的值将按通信协议要求与数据打成一个 UDP 包发送出去。目标机端收到数据包之后,将序列号提取出来和自身维护的序列号进行对比。如果两者相同,则确认此包正确,并回应宿主机,宿主机可接着发下一包;如果两者不同,则要求宿主机重发目标机所要求序列号的数据包。比如宿主机这时发送的是 0 号数据包,如果收到包时目标机维护的全局序列号是 0,则表示此包正确,目标机端将其维护的全局序列号加 1;如果收到包时目标机维护的全局序列号不是 0,则表示此包不正确,并组包要求宿主机发送自己序列号的数据包。

2.3: 通信协议设计

为了实现宿主机和目标机在下载程序时候的通信,我们借鉴了 GAISLER 公司的 EDCL 协议。EDCL 是 Ethernet Debug Communication Link 的简称,是该公司为 grlib IP core 所开发,用以通过以太网来访问片上 AHB 总线。通过该协议能够对 AHB 总线上的任意地址空间进行读写操作。

EDCL 协议要求的以太网包的格式^[5]如图 4 所示:

Ethernet Header	IP Header	UDP Header	2B OFFSET	4B Control word	4B Address	Data 0-242 4B words	Ethernet CRC
-----------------	-----------	------------	-----------	-----------------	------------	---------------------	--------------

图 4 EDCL 要求的以太网包格式

其中 4 字节的 Address 是用来表示所要操作的内存空间,4 字节的 Control word 说明了宿主机和目标机在传输数据时所遵循的控制方法,这 4 个字节的内容在目标机接受和发送数据时有所不同。

目标机接收数据时,Control word 的内容^[6]如图 5 所示:

16-bit Offset	14-bit Sequence number	1-bit R/W	10-bit Length	7-bit Unused
---------------	------------------------	-----------	---------------	--------------

图 5 接受数据时 Control word 的内容

其中 16 位的 OFFSET 是用来使应用层其余数据在内存中保持字对齐,R/W 读写为表明了要执行的是读操作还是写操作。10 位的长度位说明了要操作的数据的长度。如果 R/W 位为 1,则图 4 中的数据域就是要写的数据,如果 R/W 位为 0,则数据域中的内容为空。

当目标机回应宿主机时,Control word 的内容^[6]如图 6 所示:

16-bit Offset	14-bit Sequence number	1-bit ACK/NAK	10-bit Length	7-bit Unused
---------------	------------------------	---------------	---------------	--------------

图 6 接受数据时 Control word 的内容

当目标机接收数据的时候,14 位的序列号将和内部维护的一个计数器进行匹配,如果两者不匹配,则不进行任何操作,ACK/NAK 位置 1,回复包的序列号域为内部计数器的值;如果两者相匹配,则进行相应的操作,ACK/NAK 位置 0,内部计数器的值加 1,回复包的序列号域的为内部计数器更新后的值。当 ACK/NAK 位置 1 时,长度域的内容为 0。

由于我们不涉及到读操作,我们可以将注意力集中在 R/W 为 1 的情况。并且因为我们是在软件上模拟内部计数器,而当要传输的文件比较大时,数据包的数量可能超过 14 位序列号所表示的范围,因此在进行序列号和内部计数器的匹配操作的时候,需要对内部计数器进行掩码操作,掩码为 0x00003fff。

2.4 源代码级调试流程

我们将做好的目标机网卡监程序和目标板的初始化程序烧制在目标板的 flash 中,这样目标板在启动的时候便会自动完成初始化并执行网卡监程序,这时候在宿主机执行数据发送程序下载待调试的程序。下载完毕之后,就可以启动 ARCDU 配合 GDB 进行源代码级的调试。

3、性能测试与优化

我们编写了一个 helloworld 程序来测试,测试结果如图 7 所示:

```
(gdb) target extended-remote :8888
Remote debugging using :8888
0x00000000 in ?? ()

(gdb) info break
Num Type             Disp Enb Address      What
1 breakpoint keep y 0x0000116c in main at helloworld.c:7
2 breakpoint keep y 0x0000118c in main at helloworld.c:9

(gdb) d 2
(gdb) info break
Num Type             Disp Enb Address      What
1 breakpoint keep y 0x0000116c in main at helloworld.c:7

(gdb) c
Continuing.

Breakpoint 1, main () at helloworld.c:7
Current language: auto; currently c

(gdb) b 9
Breakpoint 3 at 0x0000118c: file helloworld.c, line 9.

(gdb) c
Continuing.

Breakpoint 3, main () at helloworld.c:9

(gdb) c
Continuing.

Program exited normally.
Current language: auto; currently asm
```

图 7 测试结果

测试结果表明,在通过以太网下载了待调试的文件之后,在 gdb 下不必执行 download 命令即可进行调试。

在测试过程中,我们发现宿主机发包程序的组包长度下载速度有着较大的影响。在测试文件大小为 5M 的情况下,包长 512 字节需时 12 秒,包长 256 字节则需时 21 秒。

4、总结

本文通过实现以太网来下载待调试程序,改进了通过串口下载程序较慢的缺点,对缩短在 arc810 平台上开发嵌入式软件的周期具有重要意义。该软件不仅能下载待调试程序,而且还能下载 linux 印象文件至开发板,软件已通过测试,在这里感谢厦门宜展科技提供的帮助。

参考文献:

1. 吴志雄,周剑扬,卢敏. 一种易于扩展的交叉调试器设计及其实现[J]. 电子技术,2007/23
2. LEON2 Processor User's Manual[Z]. <http://www.gaisler.com/>, 2005
3. ACARD 810 User's Manual[Z]. 2005
4. VIA Rhine Family Fast Ethernet Controller Programming Guide[Z], 2007
5. GRlib IP Core User's Manual[Z]. <http://www.gaisler.com/>, 2007.